User Manual

Annex 2 - Advanced use for scientific applications

	Name	Date	Signature
Prepared by	Jean-Charles Matéo-Vélez, Pierre Sarrailh and Julien Forest	30/10/2013	
Verified by	Benoit Thiébault and Benjamin Ruard	30/10/2013	
Approved by	Jean-Charles Matéo-Vélez and Julien Forest	30/10/2013	

Presentation

Objectives

The objective of this document is to provide the user with the relevant information to understand how to use some of the advanced SPIS capabilities, especially those developed in the frame of SPIS-SCIENCE activity initiated by ESA.

This UM is compliant with the version 5.0.1 of SPIS.

We remind that SPIS is an expert application. For this reason, the relevance of the results remains of the responsibility of the user.

Changes

Version	Revision	Date	Auteur / Observation
1	0	21/03/2013	Julien Forest/ Document creation
1	3	26/03/2013	Jean-Charles Matéo-Vélez / Update for numerical kernel
1	4	12/06/2013	J. Forest / update extended material properties settings
1	5	26/06/2013	J.C Matéo-Vélez / Final

Table of contents

1. Introduction

The present document constitutes an annex to the User Manual (UM) [RD0] dedicated to the use of extended SPIS capabilities for scientific applications. Many of these capabilities arose from the SPIS-SCIENCE activity (ESA contract 4000102091/10/NL/AS). Details of numerical implementation were described elsewhere [RD 10].

Section 2 and section 3 present how to generate user-defined distribution functions and user-defined surface interactions, respectively.

In section 4, the control of scientific instruments is detailed for particle detectors, Langmuir probe sweeps and for virtual particle detectors.

Section 5 describe how to generate unmeshed surfaces for diverse applications: solar panels, thin wires, semi-transparent grids and virtual surface.

Section 6 presents the settings for transient phases pre-defined scenario.

1.1. Reference documents

[RD0] SPIS User Manual, 2013

[RD1] SPINE community Web site, www.spis.org

[RD2] Artenum's Kerdiwen Web site, http://www.artenum.com/EN/Products-Keridwen.html

[RD3] SPIS services Web site, http://www.spis-services.eu/

[RD4] Apache Felix OSGI runtime Web page, http://felix.apache.org/site/index.html

[RD7] ONERA, SPIS Project Documentation, *How to control NUM from UI*, 2012.

[RD8] ONERA, SPIS Project Documentation, *How to define spacecraft equivalent circuit*, 2012.

[RD9] Cassandra Web site, http://dev.artenum.com/projects/cassandra

[RD10] ONERA, SPIS-SCIENCE Software Design Document, 2013

1.2. Acronyms, abbreviations and definitions

- **TBC**:

1.3. Caution and limit of warranty

The present document is intended to be an annex to User Manual (UM) to help the user to use SPIS capabilities. However, we remind that SPIS is still an expert tool. This document does not attempt to provide to the future SPIS user the whole expertise and knowledge needed to perform the complete modelling of a spacecraft and properly analyse the extracted data.

The final confidence in the relevance and accuracy of simulation results depends on the user expertise and remains of his responsibility.

FOR THIS REASON, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING, THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT

WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

In spite of the simplification of the user interface, it is still strongly recommended to have good knowledge in space environment, space engineering and mathematical modelling before any simulation of realistic and/or critical systems is performed.

For an improved handling of the key concept addressed by SPIS, we strongly recommend to contact ESA and ONERA space environment experts and/or follow one of the training programs proposed by ONERA and Artenum [RD3].

For further information, please contact:

- General expertise contact: contact@spis.org
- SPIS trainings offers contact@artenum.com or see www.artenum.com/services

The present document will focus on the use of the SPIS Instruments on the basis of SPIS 5.1.X.

2. User defined distributions functions

SPIS provides users with means to define their own distribution functions for ambient and secondary particles, extending the legacy Maxwellian distributions. An unlimited number of distributions can be defined through the global parameter list and using ASCII files.

For new ambient populations, several steps are necessary at global parameter level:

- environmentType must be set to **ExtendedEnvironment**. This environment permits to add new generic populations to the legacy BiMaxwellianEnvironment.
- ExtendedPopNbr is the number of populations added to available bi maxwellian electron and ion ambient populations.
- The complete set of parameters is described in *ControllingNUMfromUI.html* [RD7], including same parameters as for classical distribution and new parameters for distribution functions (DF) of PIC populations. The DF are passed or transformed internally in tabulated DF.
- In case, new populations are modelled by PIC : pop#VolDistrib must be PICVolDistrib.
- In this case, the first possibility is to define a DF from internally coded functions. Various DF can be selecting through the global parameter pop#EnvironmentDF where # is the index of the extended population
    - IsotropicMaxwellianDF is controlled with pop#Temperature
    - IsotropicBiMaxwellianDF is controlled with pop#Temperature1 and pop#Temperature2, and with pop#RatioN1overN2 and with pop# RatioJ1overJ2 which are the density and current density ratio of population 1 over population 2, respectively.
    - IsotropicKappaDF1 is the kappa 3D distribution defined by

      $$f_\kappa(v)=\frac{N}{\pi^{3/2}}\frac{1}{\theta^3}\frac{\Gamma(\kappa+1)}{\kappa^{3/2}\Gamma(\kappa-1/2)}\left(1+\frac{v^2}{\kappa\theta^2}\right)^{-(\kappa+1)},$$

      $$\theta=\left[(2\kappa-3)/\kappa\right]^{1/2}(kT/m)^{1/2}$$

      which is controlled by pop#Temperature and by pop#Kappa
    - MaxwellianDF (Maxwellian 3V distribution) which is controlled with pop#Tx, pop#Ty, pop#Tz, pop#Vx, pop#Vy, pop#Vz as defined in the natural SPIS reference basis.
- The second possibility consist in using user-defined tabulated DF. Two sub-cases are available
    - IsotropicTabulatedDistributionFunction defined by the ASCII file referenced in the global parameter pop#DF_FileName:

v (in m/s)	f(v) (arbitrary unit)
10	1
100	0.99
1000	0.95

10000 0.9

50000 0.7

…

- o TabulatedDistributionFunction which is a DF defined versus three velocity components, basically in a Cartesian coordinates system, but also possibly in polar coordinates. It requires the definition of the pop# orthogonal reference basis (Vect1, Vect2, Vect3) defined wrt to SPIS natural basis with (DFBasis_Vect1_X, DFBasis_Vect1_Y, DFBasis_Vect1_Z) and (DFBasis_Vect2_X, DFBasis_Vect2_Y, DFBasis_Vect2_Z). The DF is then described in the local basis using pop#DF_FileName. Two formats are offered:

    - mode 1: matrix format in which the ranges for Vx, Vy and Vz are indicated at the beginning of the file, then 2D slices at different values of Vx

    - mode 2: “box” format in which the phase space is separated in 3D elements of possibly different sizes. It is the responsibility of the user to ensure the full velocity space is covered.

→ TabulatedDistributionFunction (mode 1)

```
# X tab (table of X values)
-200000.0
-100000.0
0.0
200000.0
400000.0
# Y tab (table of Y values)
-5.0e+4
-2.5e+4
-1.0e+4
0.0
1.0e+4
1.5e+5
3.0e+5
# Z tab (table of Z values)
-8e+4
-4e+4
0.0
1.0e+5
1.5e+5
# X = -2000.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
0.0         0.0         0.0         0.0         0.0
# X = -1000.0
0.0         0.0         0.0         0.0         0.0
0.0         0.9         0.9         0.9         0.0
0.0         0.9         0.9         0.9         0.0
0.0         0.9         0.9         0.9         0.0
0.0         0.9         0.9         0.9         0.0
0.0         0.9         0.9         0.9         0.0
0.0         0.0         0.0         0.0         0.0
# X = 0.0
…
```

→ TabulatedDistributionFunction (mode 2)

```
# vx      vy        vy        f         dvx       dvy       dvz       df/dvx    df/dvy
          df/dvz
-100.0    -10.0     -10.0     1.0       90.0      10.0      10.0      0.0       0.0       0.0
-100.0    -10.0     10.0      1.0       90.0      10.0      -10.0     0.0       0.0       0.0
-100.0    10.0      -10.0     1.0       90.0      -10.0     10.0      0.0       0.0       0.0
-100.0    10.0      10.0      1.0       90.0      -10.0     -10.0     0.0       0.0       0.0
-10.0     -10.0     -10.0     1.0       10.0      10.0      10.0      0.0       0.0       0.0
-10.0     -10.0     10.0      1.0       10.0      10.0      -10.0     0.0       0.0       0.0
-10.0     10.0      -10.0     1.0       10.0      -10.0     10.0      0.0       0.0       0.0
-10.0     10.0      10.0      1.0       10.0      -10.0     -10.0     0.0       0.0       0.0
0.0       -10.0     -10.0     1.0       2.0       10.0      10.0      0.0       0.0       0.0
0.0       -10.0     10.0      1.0       2.0       10.0      -10.0     0.0       0.0       0.0
0.0       10.0      -10.0     1.0       2.0       -10.0     10.0      0.0       0.0       0.0
0.0       10.0      10.0      1.0       2.0       -10.0     -10.0     0.0       0.0       0.0
2.0       -10.0     -10.0     1.0       10.0      10.0      10.0      0.0       0.0       0.0
2.0       -10.0     10.0      1.0       10.0      10.0      -10.0     0.0       0.0       0.0
2.0       10.0      -10.0     1.0       10.0      -10.0     10.0      0.0       0.0       0.0
2.0       10.0      10.0      1.0       10.0      -10.0     -10.0     0.0       0.0       0.0
…
```

- Secondary electron emission under electron or proton impact is activated by pop# SEEFlag (1: activates SEE; 3: also simulates secondaries with PIC; 7: also allows secondaries of secondaries)
- Erosion is activated by pop#Erosion (1: activates erosion; 3 : also simulates products of erosion by PIC)
- Optimization of particle injection is controlled with pop#Optimization. As example pop1Optimization = 0.5 will add 50 % of injected particles in zones automatically identified by the code (with local enhancement aiming at increase statistics in the computational volume and close to spacecraft surfaces).

3. User defined spacecraft interactions

3.1. User defined distributions for secondary particles

Secondary electrons from electron, proton and photon impact can be modelled with the **isotropic distribution functions** defined in section 2 (no warrantee for 3V DF since the reference basis is applied whatever the surface of injection).

3.2. Secondary electron emission yield under electron impact (SEEE)

SEEE yield can also be tuned with extended material properties of SPIS-5, though tabulated functions. The user can add new material properties to the material properties list presented at group editor level. The flag parameters permit to activate the tabulated yields instead of the legacy analytical models.

Name	Description	Variable type	Unit	Default value
SEEEYFlag	SEE by electron yield dependence in energy flag (0-analytical model / 1-tabulated mode)	Int	[-]	0.0
SEEEYETab	SEE by electron dependence in energy tab	Array	[eV,-]	-
SEEEYTTab	SEE by electron dependence in angle tab	Array	[rad,-]	-
BCKEAFlag	Backscattering of electron albedo dependence in energy flag (0-analytical model / 1-tabulated mode)	Int	[-]	0.0
BCKEAETab	Backscattering of electron albedo dependence in energy tab	Array	[eV,-]	-
BCKEATTab	Backscattering of electron albedo dependence in angle tab	Array	[rad,-]	-

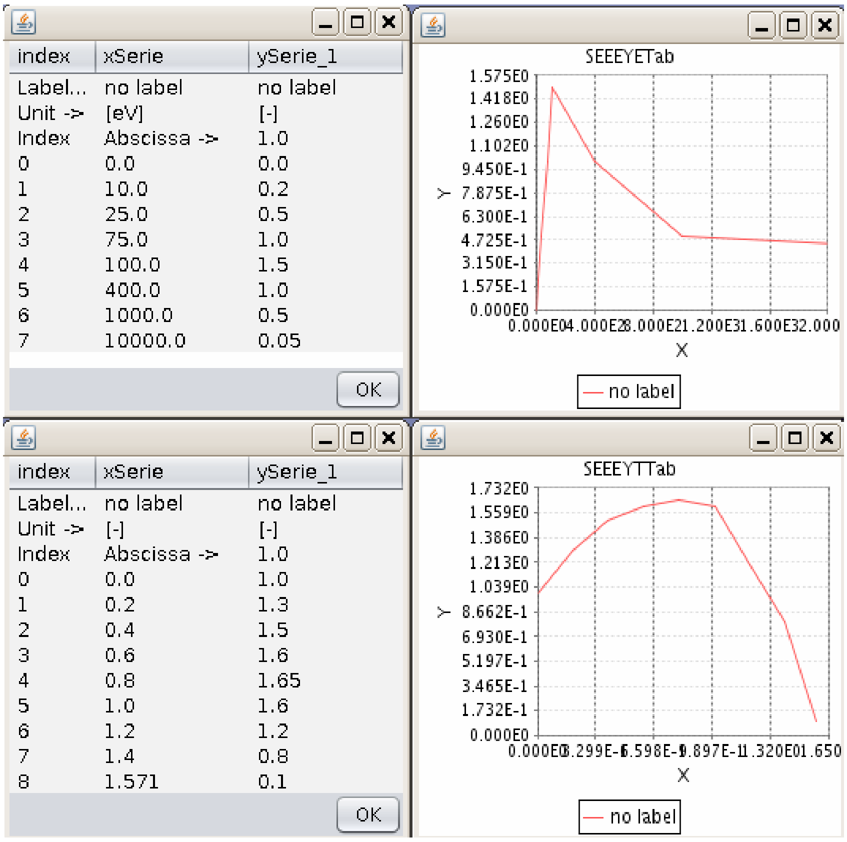


Figure 1 -Example of tabulated SEEE yield as a function of impacting electron energy (absolute value) and incidence angle (relative value)

To create new Properties, several approaches are possible. New properties files can be edited manually as all XML file. However, the simplest one is to use the advanced editor in the selection tree *Properties and Groups Editor* of the , as follow:

1. Select the group to which one the Property should be attributed to;
2. On the selected group, access to the contextual menu with a right-button click, add a new Property.
3. Select the newly created property and add, in the same way, a new Characteristics of type Series, as illustrated in Figure 2.

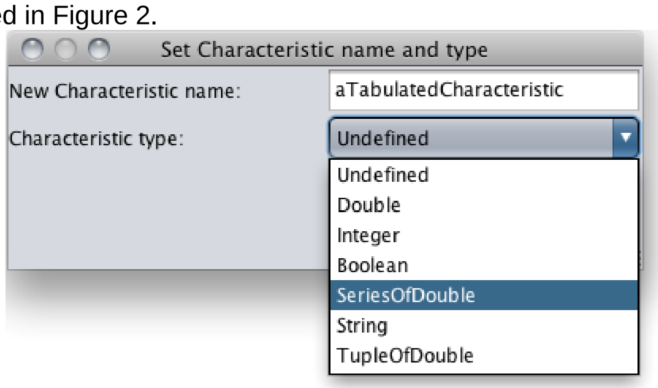
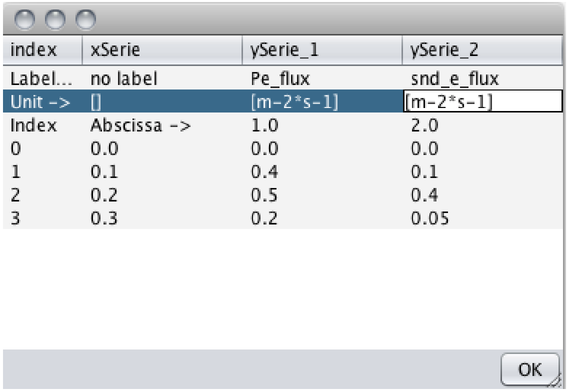


Figure 2: Creation of a new Characteristic of type series of double.

4. Set the number of series (i.e. number of columns) and the size of the series (i.e. number of values by series);
5. Click in the Edit button to set the values of series. An editor should appear as illustrated in Figure 3.



index	xSerie	ySerie_1	ySerie_2
Label...	no label	Pe_flux	snd_e_flux
Unit ->	[]	[m-2*s-1]	[m-2*s-1]
Index	Abscissa ->	1.0	2.0
0	0.0	0.0	0.0
1	0.1	0.4	0.1
2	0.2	0.5	0.4
3	0.3	0.2	0.05

Figure 3: Editor of Series Characteristics. Here two series of four values each are defined. Labels and units can be defined for each series.

6. By clicking on the plot button, series can be displayed as illustrated in Figure 4.

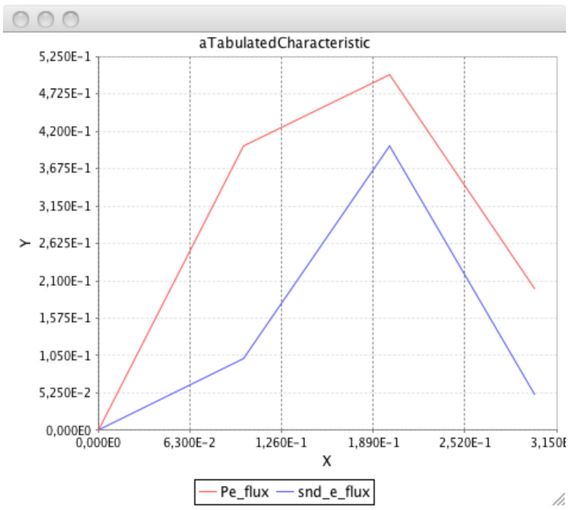


Figure 4: Example of tabulated characteristics.

7. After a complete setting of the Property and its related Characteristics, select the built Properties, with the right-click button, access to the contextual module and select the “Save this property” item, as illustrated in Figure 5.

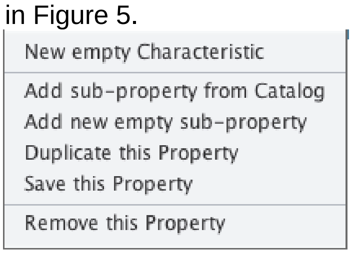


Figure 5: Contextual menu to save the selected Property.

We recommend to same all selected Property into a directory with an “.xcat” extension (should be created manually before) to constitute a new catalogue. Such directory can then be loaded as usual.

```xml
<Property>
 <id>99</id>
 <name>exampleOfRichProperty</name>
 <description>another property</description>
 <characteristicList>
  <entry>
   <string>aTabulatedCharacteristic</string>
   <SeriesOfDoubleCharacteristic>
    <id>-1</id>
    <name>aTabulatedCharacteristic</name>
    <parentProperty class="Property" reference="../../../.."/>
    <unit class="SimpleStringUnit">
     <unit>[m]</unit>
     <standardUnit>m</standardUnit>
    </unit>
    <localisation>-1</localisation>
    <informationLink></informationLink>
    <value>
     <double-array>
      <double>0.0</double>
      <double>0.1</double>
      <double>0.2</double>
      <double>0.3</double>
     </double-array>
     <double-array>
      <double>0.0</double>
      <double>0.4</double>
```

```xml
      <double>0.5</double>
      <double>0.2</double>
    </double-array>
    <double-array>
      <double>0.0</double>
      <double>0.1</double>
      <double>0.4</double>
      <double>0.05</double>
    </double-array>
  </value>
  <xSerie reference="../value/double-array"/>
  <seriesAbscissa>
    <null/>
    <double>1.0</double>
    <double>2.0</double>
  </seriesAbscissa>
  <ySeriesLabel>
    <string>no label</string>
    <string>Pe_flux</string>
    <string>snd_e_flux</string>
  </ySeriesLabel>
  <seriesUnit>
    <SimpleStringUnit>
      <unit>[]</unit>
      <standardUnit></standardUnit>
    </SimpleStringUnit>
    <SimpleStringUnit>
      <unit>[m-2*s-1]</unit>
      <standardUnit>m-2*s-1</standardUnit>
    </SimpleStringUnit>
```

```xml
      <SimpleStringUnit>
        <unit>[m-2*s-1]</unit>
        <standardUnit>m-2*s-1</standardUnit>
      </SimpleStringUnit>
    </seriesUnit>
    <size>4</size>
   </SeriesOfDoubleCharacteristic>
  </entry>
 </characteristicList>
 <isCompound>false</isCompound>
 <type></type>
</Property>
```


Table 1 - Example of saved rich Property with Series Characteristics.

4. Scientific Instruments

In the frame of the SPIS-SCI project, a set of additional monitoring and analysis tools, called Spis Instruments, has been developed. These SPIS instrument allow defining real or virtual instruments and probes inside the computational volume or on the spacecraft surface to extra and analyse finely physical data. These instruments are in complement with those described in [RD0]. The reader can refer to SUM for the general description of instruments and for details on plasma sensors and live monitoring.

We focus in this section on three instrument types:

- **Particle detectors**, in the large sense of the term, including:
    - **Particles Detector** itself provides mainly the particle distribution functions on dedicated spacecraft surfaces. They basically rely on a Test Particle (TP) method which consists in calculating the particle trajectories in a frozen electromagnetic field, by a series of forward and backward tracking.
    - **Langmuir probe** instrument extends the Particle Detector class by introducing IV sweep coupled with TP.
- **Virtual particle detector** provides the same information as particle detectors but on a surface independent from the spacecraft. They do not interact with the plasma and spacecraft dynamics. They are used to estimate currents flowing through a virtual surface immersed in the computational volume.

4.1. General configuration and use

When requesting a new instrument using the  button of the instrument wizard, the user can choose between two possibilities in the combo box :

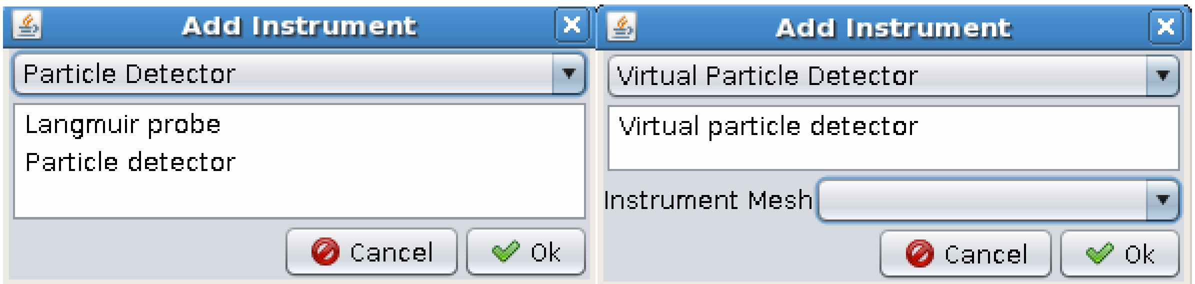
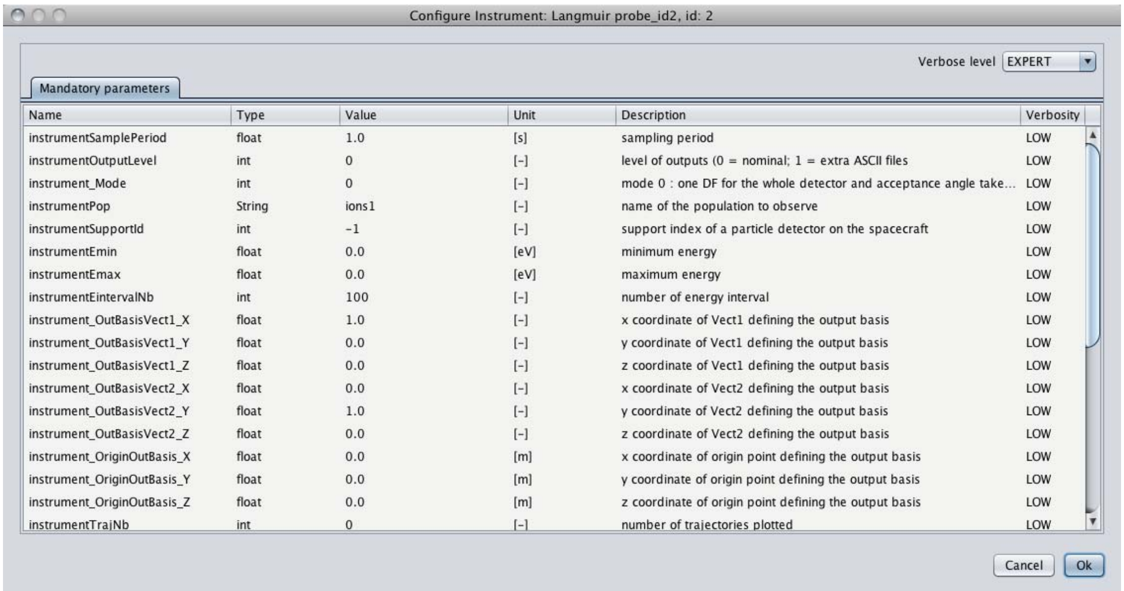


Figure 6 - Add a new particle detector, Langmuir probe or virtual particle detector

As for any instrument, an table of instrument parameters automatically appears when selecting a new one. These parameters are described in detail in next paragraphs.



Name	Type	Value	Unit	Description	Verbosity
instrumentSamplePeriod	float	1.0	[s]	sampling period	LOW
instrumentOutputLevel	int	0	[-]	level of outputs (0 = nominal; 1 = extra ASCII files	LOW
instrument_Mode	int	0	[-]	mode 0 : one DF for the whole detector and acceptance angle take...	LOW
instrumentPop	String	ions1	[-]	name of the population to observe	LOW
instrumentSupportId	int	-1	[-]	support index of a particle detector on the spacecraft	LOW
instrumentEmin	float	0.0	[eV]	minimum energy	LOW
instrumentEmax	float	0.0	[eV]	maximum energy	LOW
instrumentEintervalNb	int	100	[-]	number of energy interval	LOW
instrument_OutBasisVect1_X	float	1.0	[-]	x coordinate of Vect1 defining the output basis	LOW
instrument_OutBasisVect1_Y	float	0.0	[-]	y coordinate of Vect1 defining the output basis	LOW
instrument_OutBasisVect1_Z	float	0.0	[-]	z coordinate of Vect1 defining the output basis	LOW
instrument_OutBasisVect2_X	float	0.0	[-]	x coordinate of Vect2 defining the output basis	LOW
instrument_OutBasisVect2_Y	float	1.0	[-]	y coordinate of Vect2 defining the output basis	LOW
instrument_OutBasisVect2_Z	float	0.0	[-]	z coordinate of Vect2 defining the output basis	LOW
instrument_OriginOutBasis_X	float	0.0	[m]	x coordinate of origin point defining the output basis	LOW
instrument_OriginOutBasis_Y	float	0.0	[m]	y coordinate of origin point defining the output basis	LOW
instrument_OriginOutBasis_Z	float	0.0	[m]	z coordinate of origin point defining the output basis	LOW
instrumentTrajNb	int	0	[-]	number of trajectories plotted	LOW

Figure 7: Setting panel of Instruments. Example of Langmuir probe.

4.2. Setting of the geometry of support

Particle Detectors, Langmuir Probes and Virtual Particle Detector require geometrical supports to perform the measurements.

4.2.1. Groups of support for particle detectors and Langmuir probes

Particle Detectors and Langmuir are localised on real surfaces, i.e. on the spacecraft surface. In this case, geometrical supports are defined through the definition of corresponding groups (see Groups Editor section in [RD0]). In a first step, the user should activate the optional group property called SC Instrument support and choose a value of **InstrumentSupport**. The same value of InstrumentSupport can be used for several particle detectors and Langmuir probes. This is useful to measure different populations on the same spacecraft surface. Figure 8 shows an example of such setting. In a second step, the connection between the instruments and the support is done thanks to the **instrumentSupportId** parameter appearing in the popup wizard at instrument creation.

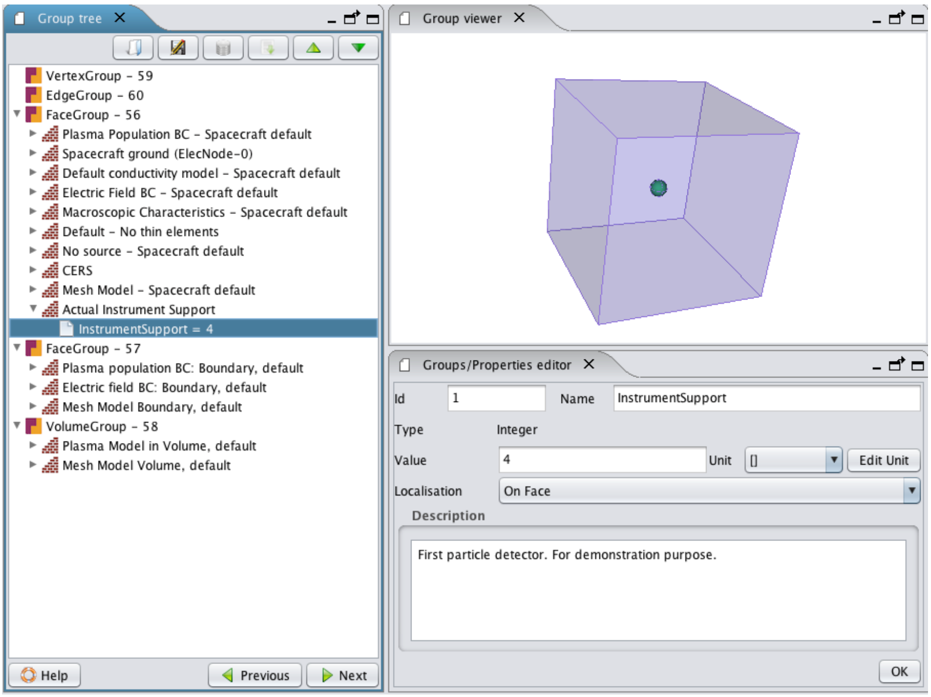


Figure 8: Setting of the geometrical support for actual particle detectors using the groups editor. Actual Particle Detector and Langmuir probes should correspond to real physical surfaces, in general a sub-part of the spacecraft surface.

See the general SPIS documentation for further information regarding the use of the Group Editor.

4.2.2. Mesh of support for virtual particle detectors

Virtual Particle Detectors require an extra mesh, called virtual mesh of support, to define the geometry of the instrument. This can be done from a CAD file, through the Geometry Editor, or directly from a mesh, through the Mesh Editor. We outline that this mesh of support is independent of the computing mesh (this is why we call it virtual). The mesh of support should be fully immersed into the computational domain as shown in Figure 9.

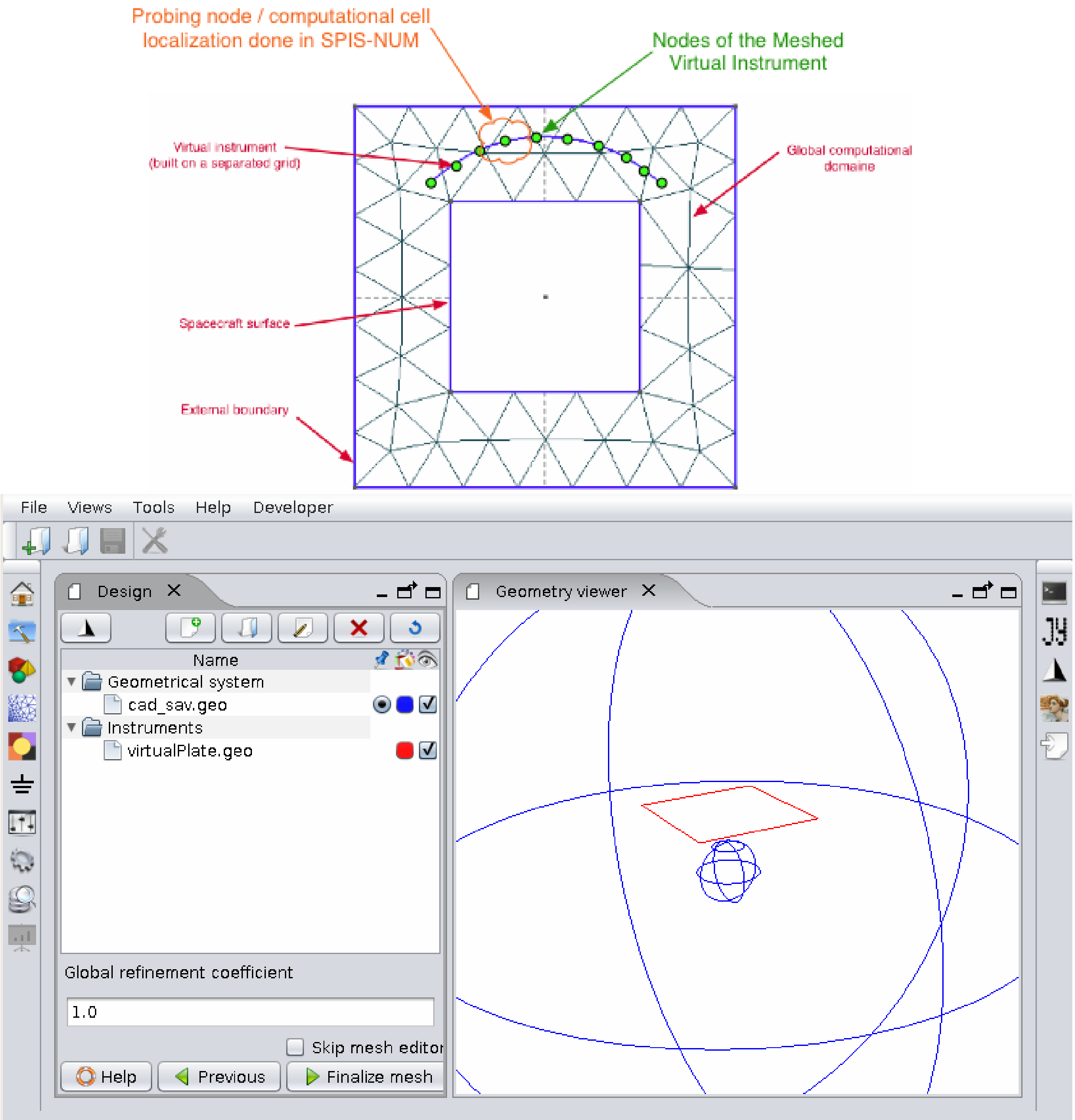


Figure 9: Top view: Definition mesh support for Virtual Particle Detector. Bottom view: Example of a virtual rectangular surface mesh above a spherical spacecraft, as defined in CAD geometry manager

The reference to the right mesh of support is set for each instrument at its instantiation as presented in next figure.

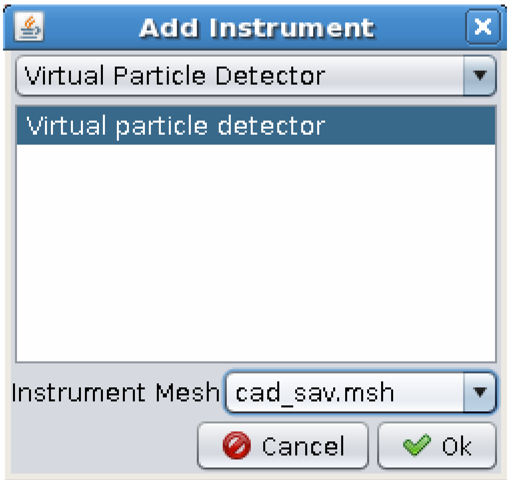


Figure 10 - selection of virtual surface mesh used by a virtual particle detector where cad_sav.msh is a mesh located at the project directory Preprocessing/Mesh/Instruments

4.3. Particle Detectors specific settings

The "*instrumentSamplingPeriod*" defines the occurrence period of a particle detector measurement in seconds. The SPIS simulation will automatically adapt the time step of the simulation to perform the measurement at the exact period the measurement is asked. When a measurement is asked by the user, the measurement period is not modified.

The "*instrumentPop*" defines the name of the population the user wants to backtrack. This population has to exist in the simulation but there are no constraints in the population type. The backtracking indifferently works for *PICVolDistrib* or *GlobalMaxwellBoltzmannVolDistrid* and for all the volume distribution available in SPIS 5. If the population selected by the user does not exist, an error message will appear as a popup or in the console at the instrument creation. At the message occurrence, the list of the population name will be displayed and the user will be invited to change the value of this parameter.

The "*instrumentSupportId*" defines the id of the surface where the instrument is plugged. It could be noticed that several instruments can hold on the same support.

The "*instrumentEmin*" and "*instrumentEmax*" defines the acceptance range in energy of the instrument. NB: this is used in the both modes thus the flux and the currents calculated by the particle detector concerns only this range of energy.

The parameter "*instrumentEintervalNbr*" corresponds to the number of interval of the plot of the results as the function of energy only in outputs.

4.3.1. Test particle

The advanced algorithm used to compute fine distribution of particle crossing particle detectors surface relies on a Test Particle approach. Particles trajectories are computed in backward mode (also forward mode during initialization) where particles are tracked from the detector surface back to original point [RD10]. The distribution is optimized thanks to an OcTree algorithm permitting to refine the 3D velocity distribution in domains of interest. The precision is controlled via two parameters : instrument_NbOctreeMax and instrument_NbPartMax. The user may carefully increase the values set by default at 10000 and 100000 keeping in mind the possible large increase in computational time and memory footprint. The selection of the population to track is done by instrumentPop.

Two modes are implemented for the particles detector. The functioning modes are selected by the parameter "*instrumentMode*".

- The mode 0 is the single distribution function mode. A single but accurate distribution function is calculated for the whole surface of the detector. The acceptance angle and the orientation of the detector are taken into account (i.e. "*instrument_OutputBasisVect*" parameters, "*instrument_LocalBasis_theta"* and "*instrument_LocalBasis_Phi*" for the orientation of the detector and "*instrument_AncceptanceAngle"* parameters for the acceptance angles). This mode is limited to a planar (or quasi-planar) surface of detection.
- The mode 1 is the multiple distribution functions mode. One distribution function is calculated for each surface elements of the detector. The statistics is lower than in mode 0 to preserve the computational costs thus the distribution function is noisy. The interest of the mode 1 is better to obtain an accurate value of the flux (not the distribution function) on a non-planar surface. The acceptance angle and the orientation are not defined in this case. This mode is used to build a Langmuir Probe for example.

4.3.2. Reference basis

The main difficulty in using particle detectors is the correct definition of reference basis. Three basis are used:

- Gmsh SPIS natural basis (X, Y, Z) defining the spacecraft geometry coordinates
- First particle detector basis (Xo,Yo,Zo) used to plot the results. This is useful to calculate the results of particle detectors in a reference frame adapted to the instrument, possibly different from the natural basis of SPIS, and possibly different from one instrument to another. It is set by the parameters instrument_OutputBasisVect. The vector coordinates are defined in the GMSH basis used by SPIS for the CAD models.
    - V1 is used as Xo
    - V2 as Yo
    - Zo is deduced from Xo and Yo to form an orthogonal direct basis
- A second particle detector basis used to define the orientation of the detector and the acceptance angles. This is useful to describe the aperture angles of instruments surfaces. Of course, instruments with different normal needs to define different basis when users want to define acceptance angles. The orientation of the detector (Xd,Yd,Zd) is defined by rotation in the (Xo,Yo,Zo) basis.
    - Rotation of θ_d around Zo to obtain the intermediary basis (X', Y', Z'=Zo)
    - Rotation of φ_d around Y' to obtain the final basis (Xd, Yd=Y', Zd)
    - The detector acceptance angles are defined in the local detector basis (Xd,Yd,Zd). The user defines 2 angles of acceptance $\pm\alpha$ and $\pm\beta$. Particles arriving within these limits are counted, others are discarded. Zd is pointing inside the particle detector surface.
        - $\pm\alpha$ around Zd in the plane (Xd, Zd)
        - $\pm\beta$ around Zd in the plane (Yd, Zd)

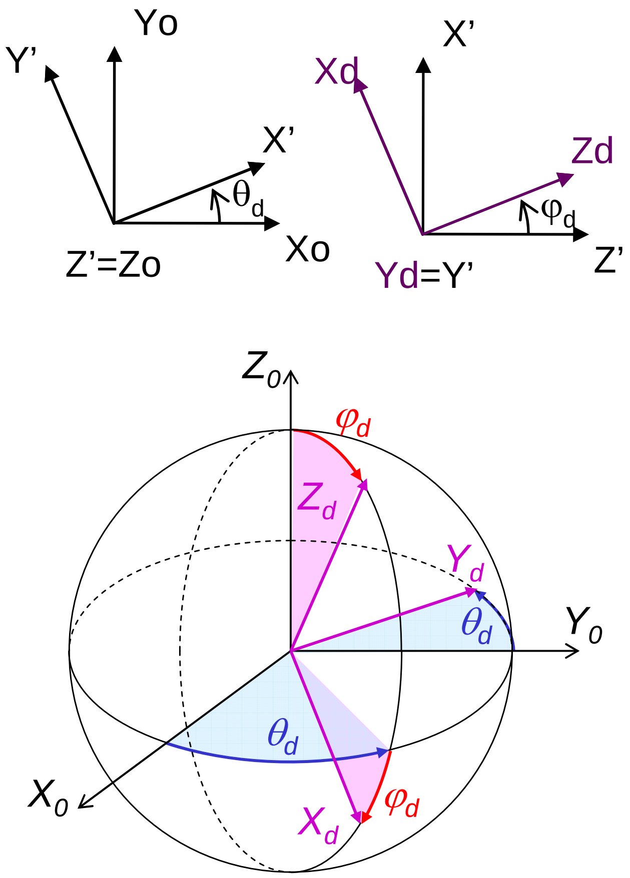


Figure 11 - Definition of the particle detector orientation

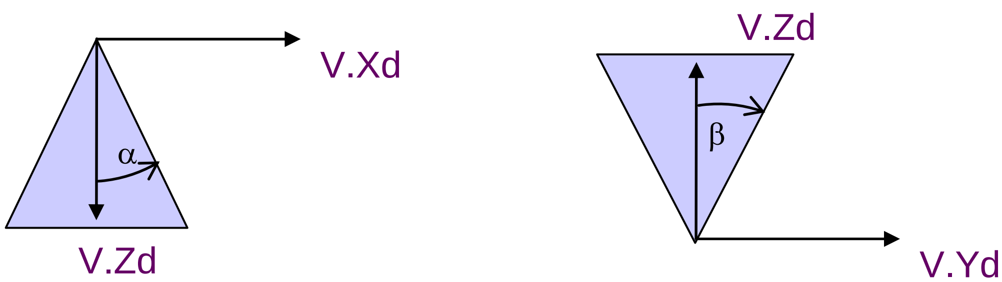


Figure 12 - Definition of acceptance angles

Definition of axis for acceptance angle

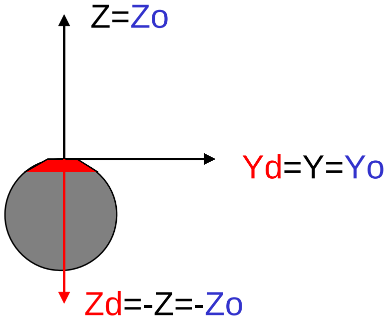


Figure 13 - Example of local basis definition for a detector (in red) located on top of a spherical spacecraft. In this case, $\theta_d = 0$ and $\varphi_d = \pi$.

Angular distributions f(θ,φ) calculated by SPIS are then given in the reference basis (Xo,Yo,Zo) by defining (θ,φ) angles:

- Rotation of θ around Zo axis
- Rotation of φ around the modified Yo axis as represented in next figure. **WARNING:** this is not the same rotation direction as for defining Zd.

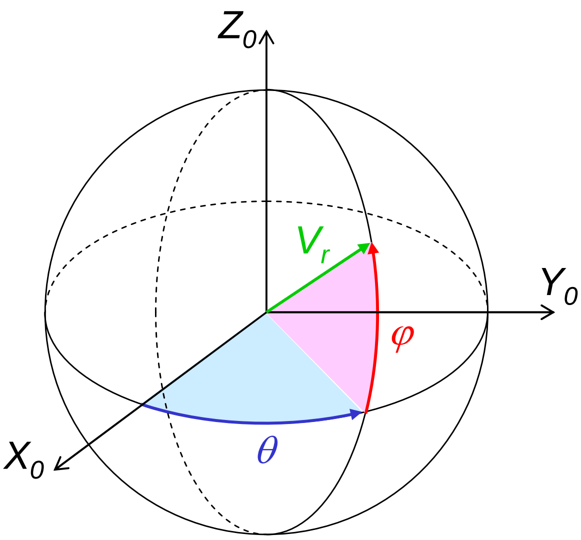


Figure 14 - Definition of angles used to monitor angular distributions

In the example corresponding to Figure 13, the angular distribution should exhibit particle collection for any θ angle but for φ angles between π and 2π approximately.

Exactly the same approach is used for virtual particle detectors. In the example below, the virtual detector aims at measuring secondary particles from the spacecraft so Zd is pointing upward.

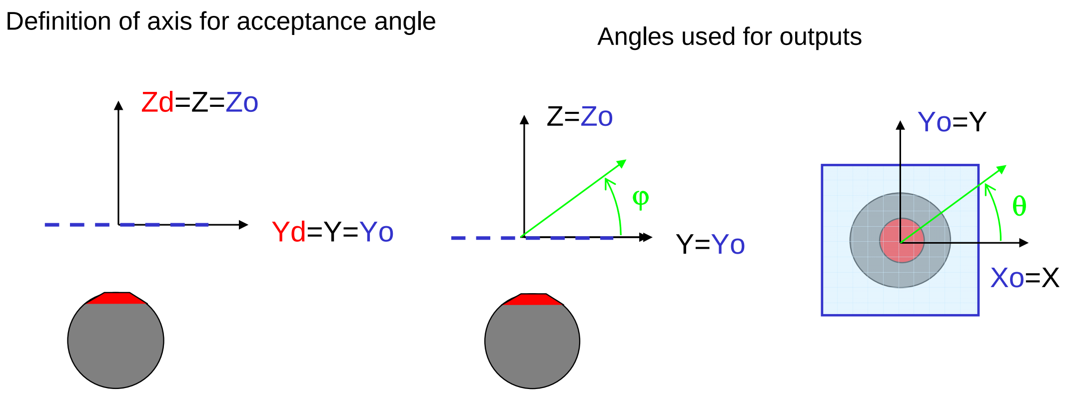


Figure 15 - Example of definition of output and local reference basis for a virtual planar detector (in blue, dashed, with transparency). In this case, θ_d = 0 and φ_d = 0.

4.3.3. Outputs

After each measurement, are automatically plotted:

- Distribution function as a function of the energy. It is performed at detection on the instrument and at injection (on boundary for ambient populations and on spacecraft for secondaries). This permits to compare the distortion of detected particles (and only detected particles)
- Differential flux as a function of the energy (at detection on instrument)
- Slice of the distribution function at a given energy as a function of elevation (phi) and azimuth (theta) angles defined in the *OutBasis* referential frame (defined at detection and injection).
- Slice of the differential flux at a given energy as a function of elevation (phi) and azimuth (theta) angles defined in the *OutBasis* referential frame (defined at detection).
- Spectrogram of the energy distribution function vs time (at detection and injection)
- Spectrogram of the differential flux as a function of the energy vs time (at detection)

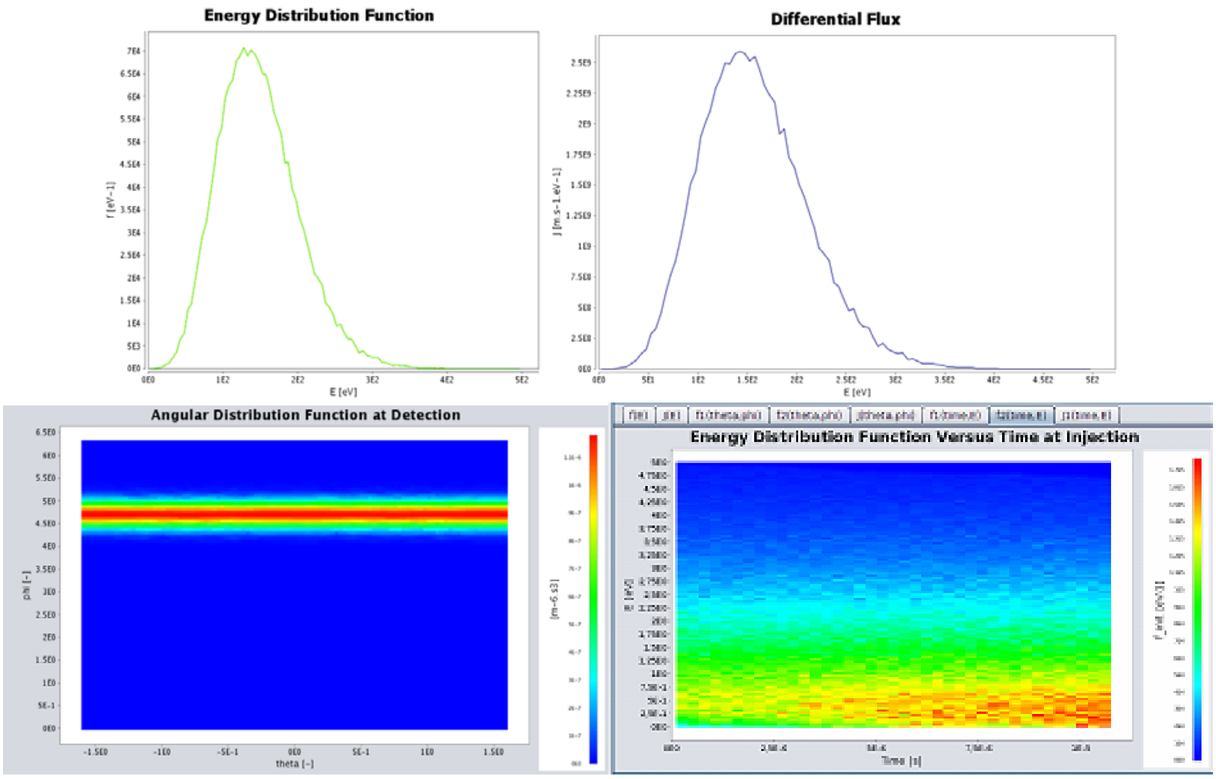


Figure 16 - Example of particle detectors outputs

In addition, 2D Data Fields (current density at detection and injection) are generated in order to clearly identify where the detected fluxes come from. This is also an **important diagnostics for scientific applications**.

The parameters "*instrumentOriginOutputs*" are used to generate the ASCII file of the particle list to define a position frame.

The "*instrument_EnergySlice*" is the energy used as reference to do the 2D slice of the angular distribution function.

The "*instrumentTrajNbr*" is the number of trajectories plotted from the detector if a sufficient number of particles are backtracked.

If the parameter "*instrumentOutputLevel*" is set to 1, ASCII files are created in NumKernel/Output directory in addition to particle measurement results:

- "spis.Util.Instrument.ParticleDetector1_Moment_at_XXXXs.txt" presents the different values of the first three moments associated to the distribution function on the detector, the flux distribution function and the distribution function at the particle source (environment or spacecraft).
- "spis.Util.Instrument.ParticleDetector1_Differential_Flux_and_Energy_at_XXXXs.txt" presents the distribution of flux, the distribution of energy and the initial distribution of energy as a function of energy.
- "spis.Util.Instrument.ParticleDetector1_Total_Current_Collection.txt" is the value of the total current on the detector as a function of time (one value by measurement).
- "spis.Util.Instrument.ParticleDetector1_2D_DifferentialFlux_at_t=XXXXs.txt" presents the 3D tabulated values of the flux distribution function on the detector in the output frame (Cartesian coordinate vx,vy,vz).

- "spis.Util.Instrument.ParticleDetector1_2D_Angular_DifferentialFlux_at_t=XXXXs.txt" presents the 3D tabulated values of the flux distribution function on the detector in the output frame (spherical coordinate E, theta, phi).
- "spis.Util.Instrument.ParticleDetector1_Velocity2DF_at_t=XXXXs.txt" presents the 3D tabulated values of the velocity distribution function on the detector in the output frame (Cartesian coordinate vx,vy,vz).
- "spis.Util.Instrument.ParticleDetector1_Angular2DF_at_t=XXXXs.txt" presents the 3D tabulated values of the velocity distribution function on the detector in the output frame (spherical coordinate E, theta, phi).
- "spis.Util.Instrument.ParticleDetector1_InitialVelocity2DF_at_t=XXXXs.txt" presents the 3D tabulated values of the velocity distribution function on the surface source of particles in the output frame (Cartesian coordinate vx,vy,vz).
- "spis.Util.Instrument.ParticleDetector1_Initial_Angular2DF_at_t=XXXXs.txt" presents the 3D tabulated values of the velocity distribution function on the surface source of particles in the output frame (spherical coordinate E, theta, phi).
- "spis.Util.Instrument.ParticleDetector1_3V_Distribution_Function_at_t=XXXXs.msh" is the 3V distribution function on the detector in the OcTree form in the GMS frame (readable in GMSH).
- "spis.Util.Instrument.ParticleDetector1_3V_Differential_Flux_at_t=XXXXs.msh" is the 3V differential flux on the detector in the OcTree form in the GMS frame (readable in GMSH).
- "spis.Util.Instrument.ParticleDetector1_3V_Initial_Distribution_Function_at_t=XXXXs.msh" is the 3V distribution function on the surface source of particles in the OcTree form in the GMS frame (readable in GMSH).
- "spis.Util.Instrument.ParticleDetector1_Particle_List_at_t=XXXXs.msh" is the list of detected particle on the detector in the output frame. There is one line per particles with successively the position on the detector (xD,yD,zD), the velocity on the detector (vxD,vyD,vzD), the flux weight of the particle on the detector (wFD), the position on the particle source (xE,yE,zE), the velocity on the particle source (vxE,vyE,vzE), the flux weight of the particle on the particle source (wFE) and the statistical weight of the particle in volume (w) which is conserved in Liouville theorem.
- map of current on the detectors in mode 1 and where the current is coming from.

4.3.4. User interactive mode

After each particle detector measurement, pending on the outputs obtained, the user may dynamically choose to:

- validate the results and resume the simulation (answer “yes”)
- or to change the instrument parameters for a new TP run. Before selecting “no, new run”, the user shall select the instrument and modify its characteristics , in order they are taken into account by the new run.

The button  is used to perform a measurement on user demand instead of waiting newt sampling time. To stop a long TP computation (which can occur when using large number of particles or OcTree), the user should select the  which automatically stops the current loop. When a particle detector is not performing a measurement, it should be possible to change its parameters, but we recommend to pause the simulation before and continue after changing the parameters in order to avoid concurrent access (with possible incoherent results or kernel crash !).

For instance, it is possible to change the energy range or the acceptance angle, as illustrated in Figure 17. This permits to reach a better statistics or to focus on selected ranges of particles. Figure 18 gives an example of such usage.

The interactive mode is available for particle-based measures only.

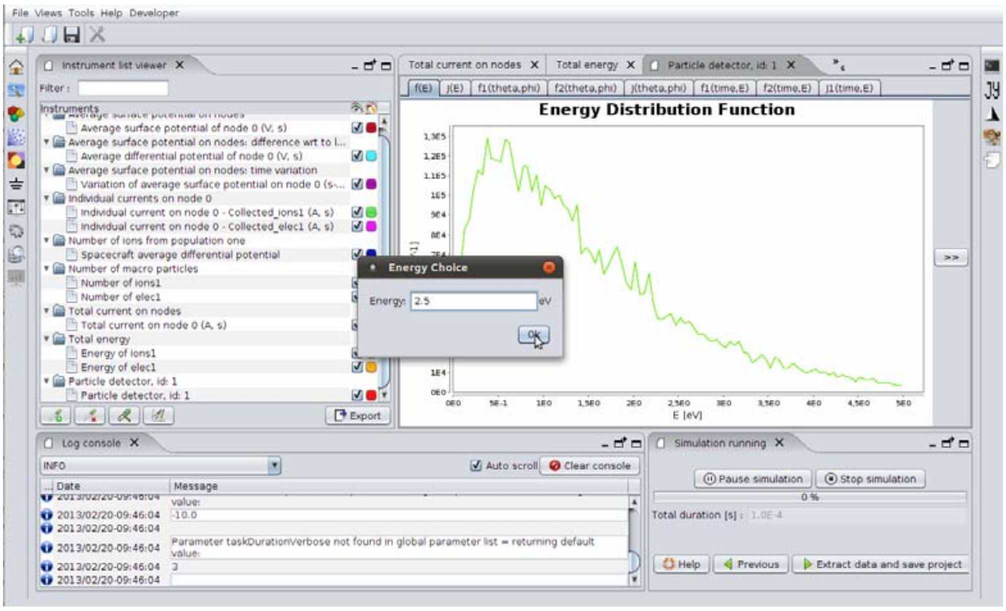


Figure 17: Modification of measure parameters in interactive mode. Here setting of the energy range of collected particles. The interactive mode is available for particle-based measures only.

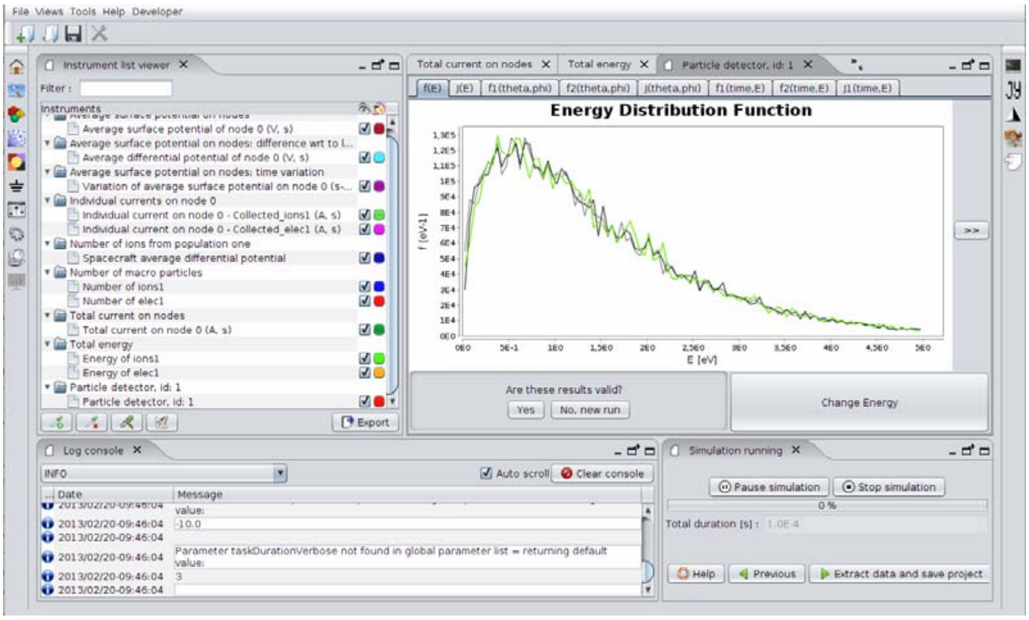


Figure 18: Interactive mode: User validation of the measure.

4.4. Virtual Particle Detectors specific settings

Apart from the geometry settings explained earlier, Virtual particle detectors have exactly the same behaviour as particle detectors.

In addition, 2D Data Fields are generated on the virtual surface mesh : potential and electric field.

4.5. Langmuir Probes specific settings

4.5.1. V sweeps

The Langmuir probe is of the same family as particle detectors. They combine the TP method with voltage sweep permitting to extract precise current-voltage characteristics on spacecraft surfaces.

As the LP is based on a Particle Detector, all the mandatory parameters of a Particle Detector are needed. **It is recommended to use the LP in mode 1** ("*instrument_Mode*" parameter).

Additional parameters are also needed in order to configure the voltage sweep Transition.

As in the Particle Detector, the "*instrumentSamplingPeriod*" defines the period of occurrence of a sequence of measurement. In the case of a LP, this period is the duration between two IV sweeps triggering (users can perform several full IV sweep at different times).

The "*instrument_DelayBetweenSteps*" defines the time delay between potential changes inside each IV sweep. It is the responsibility of the user to ensure that there is a sufficient time to do a complete IV sweep during one sampling period: "*instrumentSamplingPeriod*" > "*instrument_DelayBetweenSteps*" x ("*instrument_NbrOfSteps*" + 1)

The value of the potential range is defined with the parameters "*instrument_InitialBias*" and "*instrument_FinalBias*". The number of potential steps is defined in "*instrument_NbrOfSteps*".

If "*instrument_ReferenceElecNode*" is set to -1, the LP is in "Ground Mode". The potential of the electrical node defined in the parameter "*instrument_ElecNode*" is changed following the IV sweep. The bias is applied in comparison to the ground at infinite (0 V).

If "*instrument_ReferenceElecNode*" is set to a positive value corresponding to an existing electrical node of reference, the LP is in "Floating Mode". The potential of the electrical node defined in the parameter "*instrument_ElecNode*" is changed following the IV sweep defined with respect to the reference node.

The number of V sweep is not limited as shown in next figure.

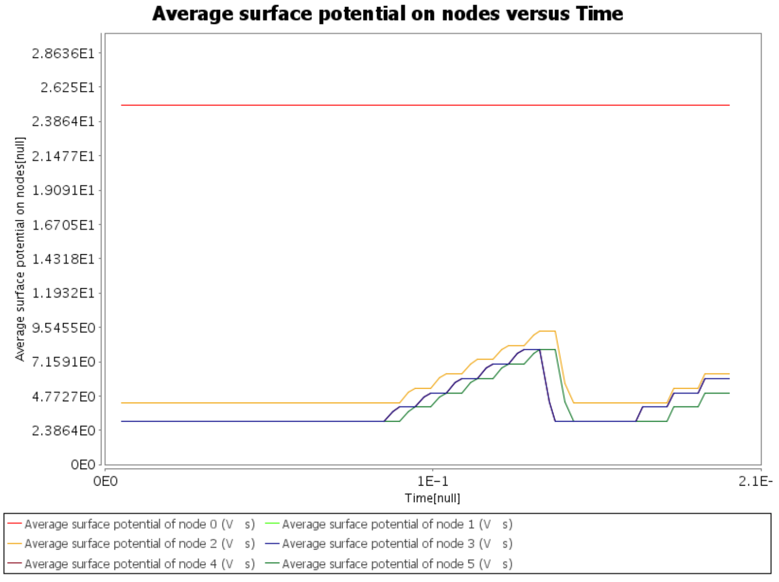


Figure 19 - Example of multiple V-sweeps

4.5.2. Two electrical configurations of LP

The LP is created to answer to two specific situations, to be chosen and set by users:

- Situation 1: Not floating spacecraft or Ground experiment
    - the circuit integration is not activated, there is no floating potential
    - there is no circuit file
    - the simulation time step should be fixed (simulationDt < 0)
    - The initial potentials are defined through globalParameters or localParameters
    - the LP sweep is defined between the infinity -1 and the LP node
- Situation 2: LP on a floating spacecraft
    - the circuit integration is activated, the potentials are floating
    - there is a circuit file
    - the simulation time is automatic (simulationDt >= 0)
    - the LP sweep is defined between two nodes => the node n1 and the LP node n2

Warning: the circuit file and the LP could have a concurrent access on the SC circuit components. As example, fixing a potential between node 0 and node 1 and between node 1 and node 2, excludes the possibility to define an IV sweep between node 0 and node 2

4.5.3. IV sweep outputs

The particle measurement using TP method is done after a delay of stabilization of the simulation between two steps of potentials (i.e. "*instrument_DelayBetweenSteps*").

First, the measurement of the current at a fixed potential is a standard output of the particle detectors. The user can use the particle detector in mode 0 or mode 1. All the standard results after a particle detector measurement are provided to the user. In addition, the current-volatge sweep is built by concatenation of current measurement at each step. An ASCII file is provided for each LP and each measurement sequence of a LP as e.g. “Langmuir probe id_IVcurve_Seq1".

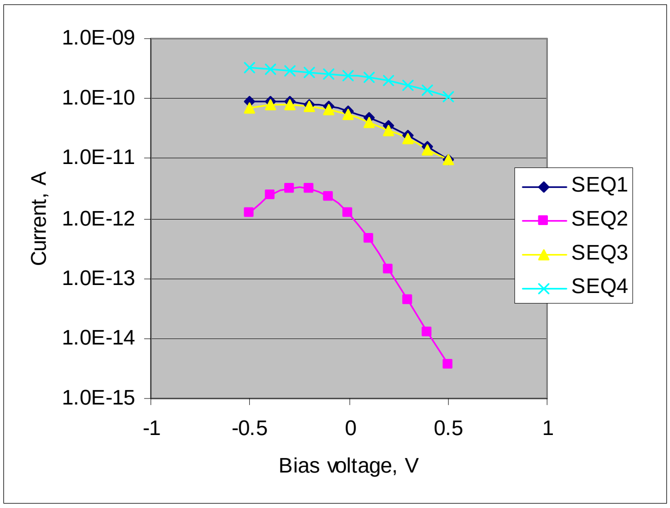


Figure 20 - Example of four sequences of I-V sweeps resulting from a Langmuir Probe submitted to changing plasma conditions (spinning spacecraft in drifting plasma). It is worth noticing the low level of noise over a large range of measured current (5 orders of magnitudes).

Remark: The test particle method is tuned when applied to LP instruments. The full velocity phase space is discretized (4π solid angle of acceptance). It aims at collecting particles on non-planar (generally spherical or cylindrical) probes.

4.5.4. User interactive mode

Langmuir probes have limited interactive capabilities as compared to particle detectors. Users can only change parameters associated to particle measurement precision (energy range, number of octree and particles, etc.) but not the voltage sweep characteristics (steps, delays, etc.).

4.5.5. Recommendations for LP design

The user is informed that some precautions must be taken when using Langmuir probes attached to spacecraft. We recommend a separation between the surface of the LPs and other surfaces. Indeed, the potential finally applied to a point at the junction of two adjacent surfaces is the average of their potential. As a result, the bias voltage may differ from what is expected.

LP sweeps should be used on elements physically disconnected from the spacecraft, as e.g. a sphere with as separation from their boom support.

4.6. Performances, memory and disk footprint

Please take care that measurements are complex operations and require to collect and process large amounts of data. For this reason, the use of an important number of instruments in parallel and/or high frequency of sampling and/or a large number of data monitored can deeply impact the whole performances of the software and, more especially, the simulation loop.

To minimise this, most of instruments performs their measure in separated threads and, in this case, a multi-core machine is strongly recommended.

Some instruments may produce a very large amount of data. Please take care to the memory and disk foot-prints.

5. Unmeshed elements

Unmeshed elements are an important improvement of SPIS 5 and especially SPIS-SCI project. 3 capabilities are available

- Solar panels : electric field singularity (described in SPIS-GEO user manual) and interconnectors

- Semi-transparent grid : metallic surface imposing a local potential to the plasma, collect a fraction of particles and emit secondaries on both sides.

NB: thin wires and solar panels as thin elements are described in SUM [RD0].

5.1. Solar panels interconnectors

Solar cells interconnectors play an important role in low temperature plasma (as e.g. Low Earth Orbit) because their bias voltage becomes sufficient to drive a large current from the plasma. The spacecraft becomes active. As it is not feasible to mesh them all, an approximate solution has been developed in SPIS.

At the length scale of the panel, we can assume that the solar array is covered by cover glasses. A fraction of the current is directly collected on the solar panel ground structure, and not on the surface of the solar cell cover glasses. The electric field is also disturbed at the vicinity of the high voltage small elements. The interconnector modelling is based on the following assumptions:

- The mesh of the solar array is defined at large scale level (not refined at interconnect level)
- The current collection on the solar array is calculated as in SPIS legacy, i.e. globally the plasma is not affected by the interconnectors
- Locally, the current is distributed between the cover-glasses and the interconnects
- The current distribution can be affected by the interconnects potentials (map defined by the user).

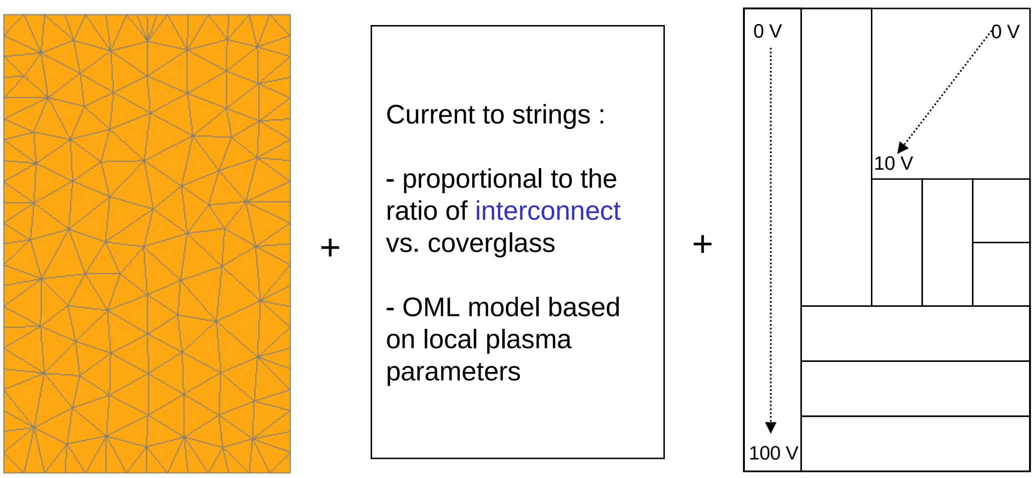


Figure 21 - Schematic representation of solar panel model. On the left : meshed surface covered with cover glass material using legacy CAD modelling. In the middle: model of current collection by unmeshed interconnectors using extra data. On the right: solar array string setup mimicking the voltages applied on solar cells and interconnectors.

The SolarArrayInteractor must be defined in different steps:

- In the global parameters list:
    - interactorNb = 1
    - interactorParticleType1 = None
    - interactorPopSource1 = All
    - interactorType1 = SolarArrayInteractor
- In the local parameters settings included in the "Plasma model type spacecraft"
    - SourceId = 1001 (i.e. 1000 + InteractorId)

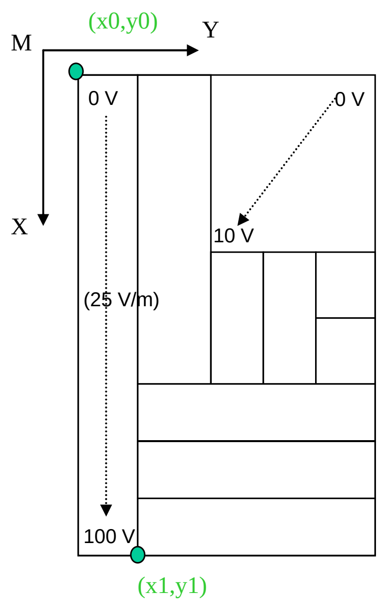


Figure 22 - Example of definition of a solar array potential map. In each zone, the potential evolves linearly

- The description files used must be located in Project -> NumKernel -> Input
    - Interactor1_SolarArrayDescriptor.txt gives the potential map description of the solar Array, see Figure 22. As example:

```
# Effective area ratio (geometric only)
1
# Basis definition
0.0     0.0     2.0     # coordinates of the ref point in SPIS natural basis
1.0     0.0     0.0     # X vector definition of the solar array basis
0.0     1.0     0.0     # Y vector definition of the solar array basis
```

Solar array definition (X0, Y0, X1, Y1, V0, dV/dX, dV/dY)

-0.5 1.0 +0.5 2.0 10 90 0

- o Interactor1_SolarArrayCollectionRation.txt gives the current ratio collected by the interconnect as a function of the relative reduced potential of the interconnect. This file is optional. In case it is missing, an OML like law is used.

Ratio of current collected by the interconnect as a function of the reduced differential potential -q(Vcoverglass- Vground)/Energy)

-100 0.0

0.0 0.0

1.0 0.5

3.0 1.0

1000 1.0

5.2. Semi-transparent grids

Semi-transparent grids (STG) are spacecraft surfaces bounded by two plasma volumes that collect only a fraction of particles passing through them. They can emit secondary particles (from electron, proton and photon impact).

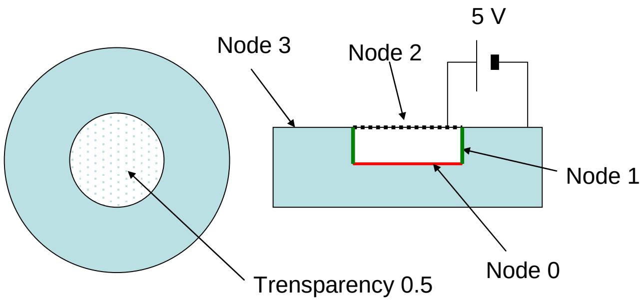


Figure 23 - Example of STG use for a the simulation of a simplified Retarding Potential Analyzer (only one grid here)

At CAD/geometry level, the STG must be defined as a surface which separates two volumes, represented by yellow spheres in Figure 24.

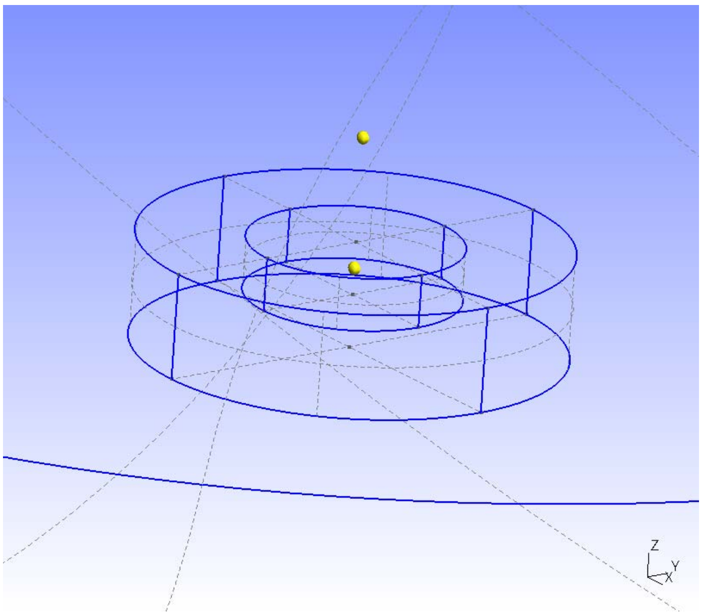


Figure 24 -CAD definition with a semi-transparent grid corresponding to Figure 23.

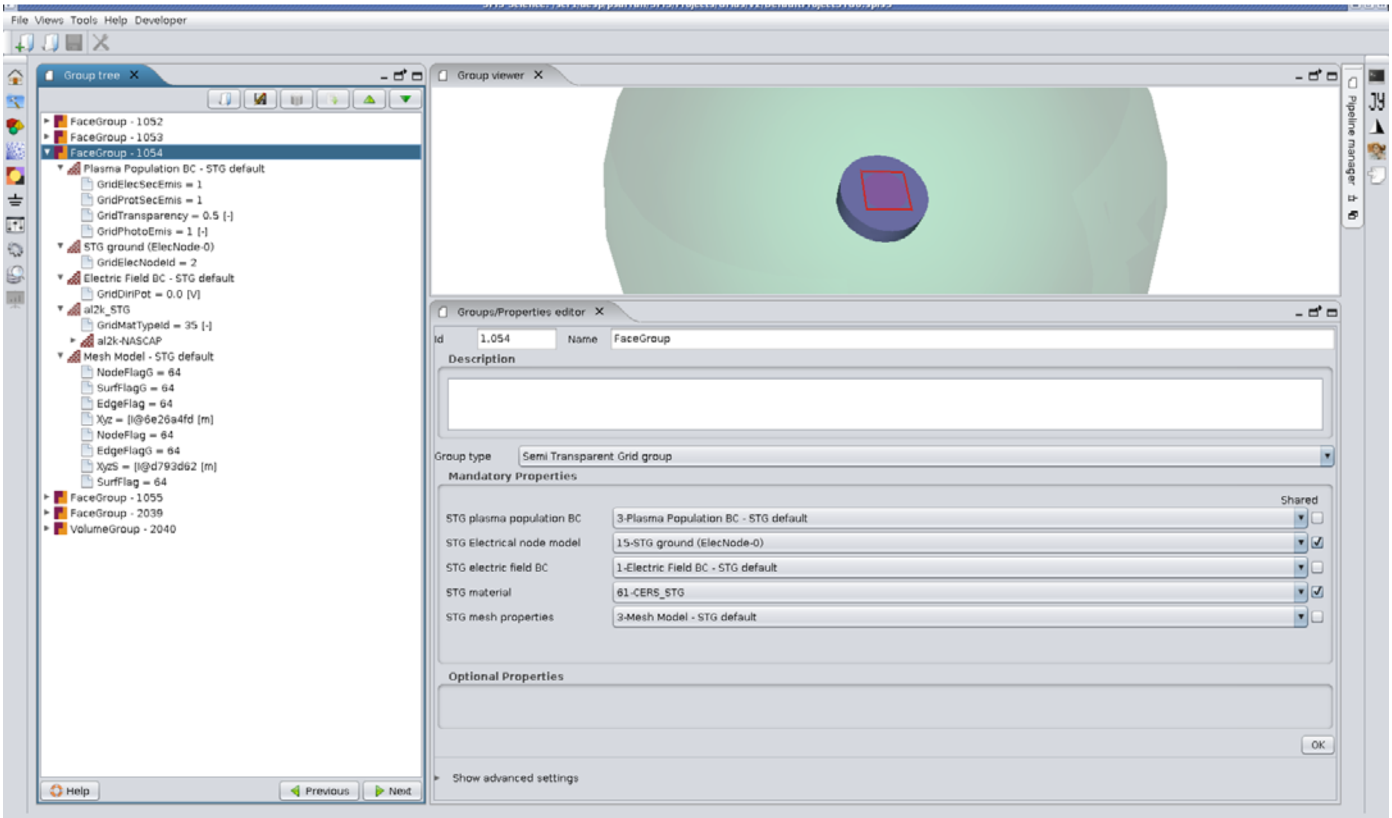


Figure 25 - Semi-transparent grid setting at group editor level

At group editor level, see Figure 25, the user defines a planar surface as “Semi-Transparent Grid” and selects:

- Grid transparency (1.0 = fully transparent)
- Electric super node linked to the grid
- Grid potential if the grid is not linked to the SC circuit
- Material on the grid (only conductors)
- Mesh model of STG (only one model)

6. Transitions

Transitions are SPIS pre-defined transient phases. The number of transitions is controlled by the global parameters of next table:

Name	Description	Variable type	Unit	Default value
transitionNb	number of transitions	int	[-]	0
transitionFlag#	flag for activating transition of index #: 0 => none, 1.0 => yes	float	[-]	0
transitionType#	name of the Transition class to be used for a transition on the simulation.	string	[-]	None
transitionDt#	maximal time step when the transition reaches a step (same role as *simulationDtInit*), permitting to perform smooth steps. The time step may automatically increase after.	float	[s]	0

As of today, user can select (and combine) the following transition types:

- SpinningSpacecraft : the rotation around an axis at a given velocity and with check time period for updating the sun flux orientation is defined in an ASCII file named SpinningSpacecraft.txt

  # Two header lines

  # only one column of data without any comment (as opposed to lines below)

  1.0 #X-Coord of spin axis

  0.0 #Y-Coord of spin axis

  0.0 #Z- Coord of spin axis

  3.1416 #SC Ang. vel. (rad/s)

  0.1 #Check time period (s)

- TransientArtificialSources: modifies the flux of a source using the ASCII file TransientArtificialSourceId.txt where sourceId is the index of the source (SourceX.Y for a sub-source)

  # Two header lines

  # Two columns : time (in second) and relative source flux wrt to initial flux defined in local parameters

  0.0 0.0

  1.0e-5 0.0

  2.0e-5 1.0

  3.0e-5 1.0

- Eclipse exit scenario used in Annex 1 is a combination of

- BasicEclipseExit: Performs a progressive change in sun flux magnitude (not its direction) as defined in BasicEclipseExit.txt:

  # Two header lines

  # Two columns : time (in second) and relative sun flux wrt to initial flux defined in global parameters

  0.0 0.0

  1000 0.0

  1100 1.0

  2000 1.0

- ConductivityEvolution: Performs a progressive change in material bulk conductivity using the tabulated BUCT material property as a function of time : BUCT [$ohm^{-1}.m^{-1}$] vs. Time [s], see section 3.2 for an illustration on how to use tabulated material properties.